Linguistics Computing Resources

Brandon Graves

Department of Ling uistics

University of Washington

September 2024

Quick Start

– Account Request Form –

https://cldb.ling.washington.edu/live/accountrequest-form.php

Contact Information

linghelp@uw.edu

CompLing Wiki

https://wiki.ling.washington.edu

Copy of this Document:

https://uakari2.ling.washington.edu/bmgraves/orientation.pdf

Web Resources

CompLing Database

Located: https://cldb.ling.washington.edu

Contains:

- Corpora we have access to and whether/where they're installed on patas.
- Linguistics Software installed on the Cluster
- Job Postings

Gitlab Server

https://git.ling.washington.edu

- You must sign up with UW email, non UW emails are blacklisted
- Accounts must be approved by an admin, this may take a few days. Email me if you have not been approved after 48 hours.
- This is a Linguistics Local resource. We do our best to maintain data integrity but having backup copies in other locations for important data is a good idea!

Welcome to Patas

Prime Directive

- **Do not Copy/paste commands from the internet.**
- **If something you need done requires “sudo”: Email me.**

Shell Access

Shell Access requires a patas account requested/approved/created.

- SSH using your UWNetID/Password:
  - Patas.ling.washington.edu
  - Dryas.ling.washington.edu
- Most Linguistics software installed under: /NLP_TOOLS
- Commonly used software(python/java/etc..) located under “/nopt”
- Corpora under: /corpora

Account request: https://cldb.ling.washington.edu/live/accountrequest-form.php

Python

- There are multiple Instances of Python installed. The system default python is not the one you want.
- Versions of python under: /nopt/ are preferable.
- Newer versions can be installed by request.
- You can create a virtual instance of python for your own use in your home directory:
  - Choose a base install of python from /nopt and use that to create your new instance
  - Ex: /nopt/python-3.6/bin/python3 -m venv ~/mypython
    - The above will create a virtual python environment called “mypython” under your home directory(~).
    - Activate that environment:
      - source ~/mypython/bin/activate

Data Access & Protection

- You can use SCP on MacOS or WinSCP on Windows to transfer files too and from patas.
- Data on patas Filesystem is backed up, but we recommend keeping your own backups as well for any important information.
- Contact me if data is deleted. I can attempt to recover it.
-

Introduction to Condor

Condor is a **batch-oriented** clustering system.

- Jobs are submit ted to a queue and matched with an available Server.
- Jobs are run non-interactively
- A “submit” file is used to tell condor how to run the job.
- Input and output are directed to files.

Unix I/O Standard

- stdin
  - Connected to the keyboard with command is run interactively.
  - Can be re-directed from a file with the < operator:
    - Mycommand <myinput.txt
- stdout
  - Connected to the screen with a command is run interactively
  - Caan be re-directed to a file with the > operator:
    - Mycommand >myoutput.txt
- stderr
  - Used for error messages and diagnostics, can be redirected separately from standard ouptut using the 2> operator.
    - Mycommand 2>error.txt

Condor has separate methods of handling all these things, but it is good to understand the overall concept.

Condor Example

Unix Command:

```
wc -w <input.txt >output.txt 2>error.txt
```

Condor Submit File

```
executable = /usr/bin/wc
getenv = true
input = input.txt
output = output.txt
error = error.txt
log = wc.log
notification = complete
arguments = “-w”
request_memory = 512
queue
```

GPU Job

GPU Nodes can be requested by adding: “request_GPUs = 1” To the Submit file

Condor Submit File

```
executable = /usr/bin/wc

getenv = true

input = input.txt

output = output.txt

error = error.txt

log = wc.log

notification = complete

arguments = “-w”

request_memory = 512

Request_GPUs = 1

queue
```

Submitting the job

- Save your job parameters to a “jobname.cmd” file.
    - Submit job with “condor_submit jobname.cmd”
- Condor adds job to queue
- When a matching machine is available, the job is executed there.
- User is notified via email when job completes.
    - username@u.washington.edu by default
    - Add: “notify_user = email@youremail.com” To send to alternate location.
    -

Job Requirements

- Condor allows you to specify how much memory your job needs
  - Use “request_memory” option in your submit file.
  - The value is interpreted in megabytes
  - Default is 2048 megabytes (2 Gygabytes)
  - If you guess low your job may be evicted; if you guess high you needlessly limited which machines can run your job.
  - The “SIZE” column in “condor_q” shows how much memory your job is currently using. You can use this to gauge your requirements for future jobs.

Advanced Options

Condor submissions have many advanced options that can allow very specific scenarios to be achieved notably:

- Splitting one job into multiple jobs to allow parallel execution across multiple servers.

- These are more naunced aspects of the compute environment but examples can be found in the condor documentation:

Visit: https://htcondor.readthedocs.io/en/latest/users-manual/submitting-a-job.html

Research Job Tracking

- We track the percentage of the cluster used by research jobs to help qualify our program for various research exemptions and grants.

- To help add "+Research=true" to your submit fiile when you run research related jobs. Do not use this for classwork.

- This does not affect job scheduling; it is for recordkeeping only.

Standard Condor Commands

- condor_submit — submits a job to the cluster
- condor_status — lists available nodes and their status
- condor_q — list the job queue
- condor_rm — remove a job from the queue

Condor Troubleshooting

Sometimes things will go wrong and jobs will either not run, or not produce expected results. When troubleshooting the following is helpful

- Check the job log file to see if there are obvious reasons it failed
- If a job is stuck in the queue use:
  - condor_q -analyze [jobid] to see why your job isn't being matched with a node.
  - condor_q -long [jobid] to see what the holdreason parameter is
- Double-check arguments and input files. Run the executable on the commandline to test for immediate failures.
- Check that all files that need to be executable have the executable flag in the filesystem.
- Check your path. Condor does not always properly translate your own filepath to its job parameters. It is best to use absolute paths for all commands and files to make sure your job runs using the files you think it is using.
- If all else fails email me at **linghelp@uw.edu,** With:
  - what errors you are seeing
  - The jobID of the problematic job
  - The Directory your submit file is located.